AUBURN UNIVERSITY

AEROSPACE

ENGINEERING

STUDY OF EFFECTS OF UNCERTAINTIES ON COMET AND ASTEROID ENCOUNTER AND CONTACT GUIDANCE REQUIREMENTS

FINAL REPORT

PART I. GUIDANCE AND NAVIGATION STUDIES

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER

Under Contract NAS8-27664

December 15, 1973

ENGINEERING EXPERIMENT STATION
AUBURN UNIVERSITY
AUBURN, ALABAMA 36830

Aerospace Engineering Department
Auburn University
Auburn, Alabama 36830

STUDY OF EFFECTS OF UNCERTAINTIES ON COMET AND ASTEROID ENCOUNTER AND CONTACT GUIDANCE REQUIREMENTS

FINAL REPORT

PART I. GUIDANCE AND NAVIGATION STUDIES

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER

Under Contract NAS8-27664

December 15, 1973

Arthur G. Bennett
Arthur G. Bennett, Associate Professor
Project Director

Robert G. Pitts
Robert G. Pitts, Professor and Head
Department of Aerospace Engineering

STUDY OF EFFECTS OF UNCERTAINTIES ON COMET
AND ASTEROID ENCOUNTER AND CONTACT
GUIDANCE REQUIREMENTS

NAS8-27664

PROJECT SUMMARY

At the start of our work in June 1971, three principal tasks were assigned. Slightly restated, these tasks were:

(i) Develop a deterministic algorithm for guidance to rendezvous with comets and asteroids that can handle expected large target ephemeris errors.

(ii) Define the problem of determination of rotational state of a tumbling asteroid or cometary nucleus and develop possible schemes for this determination.

(iii) Investigate possible contact rendezvous schemes including the "harpoon" technique.

During the first year, a detailed investigation of a rendezvous guidance technique based on "encounter theory" was conducted. The definition and formulation of the tumbling problem was made and several possible algorithms phrased. A first investigation of the harpoon problem was conducted and frequencies and acceleration levels identified.

Early in the second year work, a successful deterministic rendezvous guidance algorithm based on optimal control theory was developed. The altorithm was considered sufficiently important that, with agreement of NASA, more emphasis was placed on the rendezvous investigation. To accommodate this work, the harpoon study was set aside. An effort was initiated on the rendezvous navigation problem wherein measurements are made and statistically processed onboard the spacecraft to provide the relative state information required for input to the guidance algorithm. Expenditures on the contract were low enough so that in June 1972 a no-cost extension of the work to September 1973 was possible. At this time the changes in objective were formalized and principal tasks were restated to include the navigation work (and eliminate the contact-rendezvous and harpoon investigation).

In September 1973, delays caused by installation of a new computing machine at the University prevented generation of final data. The contract completion date was again extended, at no cost, to December 15, 1973, to allow time for this data generation and report preparation.

The final report of our work is presented in two volumes:

Part I. Guidance and Navigation Studies

Part II. Tumbling Problem Studies

Each of these volumes presents the technical details of the analyses conducted, the principal conclusions made, and listing of the computer programs employed, including descriptions of the operation of the programs. Technical abstracts of the work are included in each volume. In Part I, the body of the report reproduces a paper prepared for the AIAA 10th Electric Propulsion Conference entitled "Solar Electric Propulsion for Terminal Flight to Rendezvous with Comets and Asteroids." (AIAA Paper No. 73-1062). The title was changed for inclusion in this report and a few typographical errors were corrected.

STUDY OF EFFECTS OF UNCERTAINTIES ON COMET AND ASTEROID ENCOUNTER AND CONTACT GUIDANCE REQUIREMENTS

TABLE OF CONTENTS

STUDY OF EFFECTS OF UNCERTAINTIES ON COMET AND ASTEROID ENCOUNTER AND CONTACT GUIDANCE REQUIREMENTS

PART I. GUIDANCE AND NAVIGATION STUDIES

Abstract

A guidance algorithm that provides precise rendezvous in the deterministic case while requiring only relative state information is developed. A navigation scheme employing only onboard relative measurements is built around a Kalman filter set in measurement coordinates. The overall guidance and navigation procedure is evaluated in the face of measurement errors by a detailed numerical simulation. Results indicate that onboard guidance and navigation for the terminal phase of rendezvous is possible with reasonable limits on measurement errors.

I. Introduction

Solar electric propulsion (SEP) has been identified as the most promising means of attaining the high dynamical energies that are required for missions to comets and asteroids. However, the continuous nature of SEP complicates trajectory dynamics and introduces difficulties in guidance and navigation of the spacecraft. Handling these difficulties will require formulations that are different from those used for the impulsively corrected ballistic trajectories employed for planetary missions to date. Errors in SEP level and direction will be an important concern also because the errors occur over the whole trajectory and will be difficult or impossible to determine in advance or by direct measurement during flight.

Another problem is that the ephemerides of asteroids, and especially comets, are not as well known as those for the planets. Advance knowledge of the position of a comet such as Encke may be in error by 100,000 km. or more. One reason for this is that comets and asteroids are not observed over their full orbits or in as much detail as are planetary bodies. And the orbits of comets and asteroids of interest are more complex than those of the planets because of perturbative effects along their extended, eccentric orbits. Also, comets are acted upon by non-gravitational solar radiation pressure and electromagnetic force.

The spacecraft may proceed along a nominal path for many months or a few years and when the target can be viewed, it is found that the target is not where it is expected to be. Assuming sufficient control authority to correct for this error, there is then the navigation problem of determining the relative position and velocity of the spacecraft and target for the generation of terminal guidance commands. While transponders allow the spacecraft to be tracked from the earth with good accuracy, the accuracy of earth-based tracking of the target is much less accurate and relative state cannot be determined with sufficient precision by differencing such earth-based data. It will be necessary to make relative measurements of some kind from onboard the spacecraft to insure successful rendezvous.

So, a central problem is determination of types of onboard measurements that can or need be made. But, this problem is tied to the details of the guidance and navigation algorithms to be used and the two questions must be handled together.

Our investigations have led to a terminal guidance algorithm that gives accurate rendezvous in the deterministic case while employing a knowledge of relative state only. With sufficiently accurate relative state estimation and control, rendezvous is possible without use of ground based measurements. And if onboard systems that fit accuracy, weight, power, and cost requirements are available, a fully autonomous guidance and navigation system is possible. Such a system would eliminate signal delay for deep space targets or time-critical terminal maneuvers and relieve a heavy work load for ground based tracking systems. Very frequent measurements would be available and the accuracy of attainment of terminal conditions improved. There is, of course, the important question of availability of necessary onboard measurement and computation systems. Preliminary considerations indicate that required equipment is probably within the capabilities of present technology. A more definitive answer to this question can be given after evaluation of possible guidance and navigation schemes has been made and specific system requirements identified.

The objective of this paper is to present an approach to the onboard terminal guidance and navigation problem and some first results that indicate the approach does not require unreasonable onboard equipment.

II. The Guidance Algorithm

Simplicity is a first criteria for an onboard guidance scheme. And second, for comet and asteroid missions, the scheme must be broadly adaptable to off-nominal situations because of expected ephemeris errors. It would be desirable if the scheme is not based on linearization about a nominal, but could proceed from any state point to the desired terminal conditions.

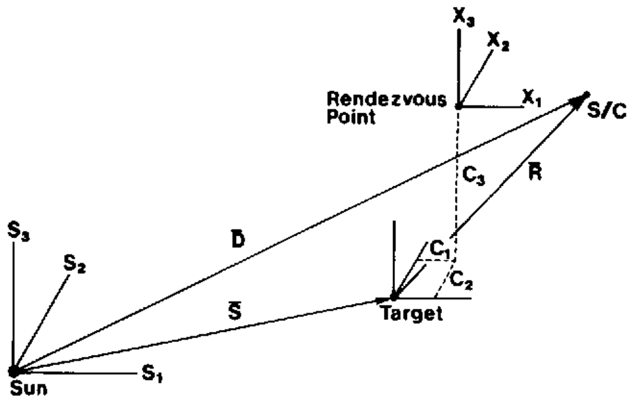

Fig. 1. Rendezvous Geometry.

The geometry and variables used to phrase the rendezvous problem are defined in Figure 1. The mass of most asteroids and certainly of comets is so small that it can be neglected in approach to rendezvous. Appropriate to onboard guidance, the equations of motion are set in coordinates relative to the target:

$$\ddot{\bar{R}} = \bar{F} + GM \left(\frac{\bar{S}}{S^3} - \frac{\bar{D}}{D^3} \right) \qquad (1)$$

where $\bar{F}$ is the SEP thrust acceleration and G is the universal gravitational constant. The problem is to determine $\bar{F}$ given the state of the spacecraft and target. An important simplification would be had if $\bar{F}$ could be specified knowing only the relative state of spacecraft and target. Of course, the dynamics of the problem are not fully specified by just relative state, but as will be found, this simplification is possible in a practical sense. The size of the sun's gradient effect as expressed in the second term on the right of Eq. (1) is the critical factor. In general, this term is larger the farther the spacecraft is from the target and the closer the target is to the sun. If we consider a target position of about 1 to 1.5 a.u. from the sun and a spacecraft position of 1.5×10^6 to 2.0×10^6 km. from the target at rendezvous, then the gradient force is of order 10^{-5} to 10^{-6} g's. Expected maximum SEP levels are about 10^{-4} g's. So, it is not unreasonable to ignore the gradient effect as a first approximation. The gradient effect could, of course, be included as a thrust separately calculated from approximate knowledge of the target ephemeris. In either case, the guidance problem reduces to free space and can be handled by a number of available mathematical approaches. These ideas are not new, having been previously employed by Cherry for lunar orbits.[3]

Written out in rectangular coordinates centered at the rendezvous point, the equations of motion are

$$\begin{aligned}
\dot{X}_1 &= X_4 \\
\dot{X}_2 &= X_5 \\
\dot{X}_3 &= X_6 \\
\dot{X}_4 &= F_1 + \frac{GM}{S^3}\left[S_1 - D_1 (S/D)^3\right] \qquad (2) \\
\dot{X}_5 &= F_2 + \frac{GM}{S^3}\left[S_2 - D_2 (S/D)^3\right] \\
\dot{X}_6 &= F_3 + \frac{GM}{S^3}\left[S_3 - D_3 (S/D)^3\right]
\end{aligned}$$

where the subscripts indicate components in the corresponding coordinate directions. M is the mass of the sun. In the spirit of approximation discussed above, the gravitational gradient terms on the right of the last three of Eqs. (2) are dropped. The remaining equations are linear, describing just a free space motion under the action of controls F_1, F_2, F_3. With SEP thrust, a most meaningful criteria for choosing these controls is to minimize their time integrated square.

$$J = \int_{T_o}^{T_f} (F_1^2 + F_2^2 + F_3^2)\, dt = \text{minimum} \qquad (3)$$

With this starting point, the solution was determined by Abercrombie.[4] The method was essentially the same as employed by several others on related problems, and is a standard procedure from optimal control theory.[5] The result is

$$\begin{aligned}
F_1 &= \left[\frac{6}{\tau_o^2}\left(1 - \frac{2\tau}{\tau_o}\right)\right] X_{10} + \left[\frac{2}{\tau_o}\left(1 - \frac{3\tau}{\tau_o}\right)\right] X_{40} \\
F_2 &= \left[\frac{6}{\tau_o^2}\left(1 - \frac{2\tau}{\tau_o}\right)\right] X_{20} + \left[\frac{2}{\tau_o}\left(1 - \frac{3\tau}{\tau_o}\right)\right] X_{50} \qquad (4) \\
F_3 &= \left[\frac{6}{\tau_o^2}\left(1 - \frac{2\tau}{\tau_o}\right)\right] X_{30} + \left[\frac{2}{\tau_o}\left(1 - \frac{3\tau}{\tau_o}\right)\right] X_{60}
\end{aligned}$$

where the X_{10}, etc., are the initial conditions and $\tau = T_f - T$ is the time to go. Note that the thrust components vary in a simple linear way with time.

The guidance algorithm expressed in Eq. (4) was evaluated with double precision digital simulation. Assuming a deterministic situation in which state is known initially, the full equations of motion, Eqs. (2), were integrated for a period (the guidance interval) with controls specified by Eq. (4). Because of the approximations made in obtaining the free space equations, the true state differs from the free space solution. At the end of the guidance interval, perfect navigation is assumed, the algorithm is updated with the new, true state, and another period flown. This procedure is repeated until rendezvous is obtained or the path diverges. Simulations were run for typical missions to comets Encke, D'Arrest, and Kopff. Initial conditions were taken from mission studies made by Friedlander.(6) Update periods from continuous to 5 days were employed. The singularity in the algorithm when time to go becomes zero was avoided by pushing time to go ahead two guidance intervals at the end of the trajectory. The time to go push was continued after rendezvous in the case of Encke to examine station-keeping properties of the algorithm. The initial conditions were varied over ranges of 10 degrees in direction of relative velocity, 10 percent in magnitude of relative velocity, and 100,000 km. in position to represent initial target ephemeris errors.

In no case was divergence of the trajectory found. The trajectories were essentially straight line approaches to the rendezvous point. Because of the crude time to go push, there were overshoots of the rendezvous points, but accurate rendezvous was obtained within one or two days of the prespecified time (40 days). In later computer runs, the overshoot has been completely eliminated by reduction of the guidance interval to 0.1 day and the time to go push to one such guidance interval. The spacecraft then performed small oscillations about the rendezvous point that increased slightly as target periapsis was approached and decreased after periapsis. For Encke, with a guidance update of 0.5 days, an overshoot of about 40 km. was obtained for a rendezvous standoff distance of 100 km. in each coordinate direction. After 40 days from guidance initiation, slow oscillations of a few meters amplitude near the rendezvous point occurred. The amplitude increased to somewhat less than 500 meters near periapsis at 100 days and decreased to a few meters by 165 days. Better time to go management would improve these results. The thrust levels during station keeping were extremely low; smaller than 0.5×10^{-8} g's. Fig. 2 shows typical thrust histories for the approach phase for the three comets. It was concluded that the free space optimal control algorithm performed well enough deterministically to warrant investigation with simulation of a realistic navigation scheme.

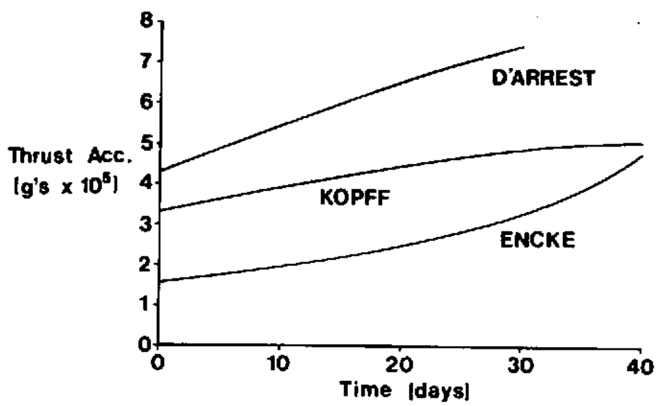

Fig. 2. Rendezvous thrust histories.

III. Navigation Scheme

A Kalman filter was chosen for forming state estimates from onboard measurements. Among the difficulties of practical use of the Kalman filter are two problems that arise because of the linear nature of the Kalman formulation. First, linearization of the system equations leads to a modeling error and it is possible that the first estimate of a new state based on linear propagation from a previous state may not be good enough for the procedure to find a good statistical correction to the first estimate. Second, the formulation requires a linear relation between the physical quantities measured and the system state. Such linear relation is not the usual case and direct linearization of the true relations can lead to serious errors especially when state is not close to estimated values. Either of these problems can produce unsatisfactory performance or divergence of the filter. The problems have been discussed extensively in the literature and there are many ways of handling them.(7,8) We will focus on an approach that is simple from the computational point of view.

The question of modeling error was first examined. A filter in which the measurements were taken as the state variables themselves was programmed for digital computation. Numerical simulation exhibited strong divergence. This was attributed to the usual fact that the filter rapidly reduces the state covariances and thus essentially ignores new measurements as they are made. At this point, one standard fix is to insert noise into the basic system equations. We took an even simpler approach and froze the state covariance matrix after 5 to 10 guidance cycles. This technique worked extremely well. Starting with gross initial errors of 10 percent of the relative state elements (200,000 km. and 150 m/sec.), rendezvous was attained with position errors of about 5 km. and velocity errors of about 0.1 m/sec. This was accomplished with the previous, crude, two-step time-to-go push. We concluded modeling error would be manageable even if requiring a more sophisticated approach in a final formulation.

A method of Mahra(9) was chosen to handle the state-measurement relation problem. In his approach, the filtering process is carried out directly in what we term measurement variables. The system state is transformed from the rectangular coordinates used for the guidance problem to new variables, some of which are the measurement quantities themselves. A one-to-one relation between state description and measurements then exists. However, this approach does, as we will see, introduce certain other approximations into the statistics, but, as results show, these are not critical.

The measurement variables vector is taken as $y = (R\ u\ v\ \dot{R}\ \dot{u}\ \dot{v})^T$, where u and v are the direction cosines for X_1 and X_2. The transformation from state variables $y = g(X)$ is

$$
\begin{aligned}
R &= [(X_1+C_1) + (X_2+C_2) + (X_3+C_3)]^{\frac{1}{2}} \\
u &= (X_1+C_1)/R \\
v &= (X_2+C_2)/R \\
\dot{R} &= [(X_1+C_1)X_4 + (X_2+C_2)X_5 + (X_3+C_3)X_6]/R \\
\dot{u} &= (X_4-\dot{R}u)/R \\
\dot{v} &= (X_5-\dot{R}v)/R
\end{aligned}
\tag{4}
$$

The inverse transformation $X = \ell(y)$ is

$$
\begin{aligned}
X_1 &= uR - C_1 \\
X_2 &= vR - C_2 \\
X_3 &= wR - C_3 \\
X_4 &= R\dot{u} + \dot{R}u \\
X_5 &= R\dot{v} + \dot{R}v \\
X_6 &= R\dot{w} + \dot{R}w
\end{aligned}
\tag{5}
$$

where, $w^2 = 1 - u^2 - v^2$

Any desirable subset of y can be chosen as the actual measurements: range R; direction cosines u and v; range rate $\dot{R}$, etc. The filter process then proceeds as follows. Starting with a best state estimate $\hat{X}_k$ at time T_k, a first estimate $X^\dagger$ at time T_{k+1} is formed by a linear extrapolation through the state transition matrix ϕ for the linearized system

$$X^\dagger_{k+1} = \phi_k \hat{X}_k \tag{6}$$

where,

$$
\phi_k = \begin{bmatrix}
\alpha_k & 0 & 0 & \beta_k & 0 & 0 \\
0 & \alpha_k & 0 & 0 & \beta_k & 0 \\
0 & 0 & \alpha_k & 0 & 0 & \beta_k \\
\gamma_k & 0 & 0 & \delta_u & 0 & 0 \\
0 & \gamma_k & 0 & 0 & \delta_k & 0 \\
0 & 0 & \gamma_k & 0 & 0 & \delta_k
\end{bmatrix}
\tag{7}
$$

and

$$
\begin{aligned}
\alpha_k &= \frac{\tau_{k+1}}{\tau_k}\left[3\left(\frac{\tau_{k+1}}{\tau_k}\right) - 2\left(\frac{\tau_{k+1}}{\tau_k}\right)^2\right] \\
\beta_k &= \tau_{k+1}\left[\left(\frac{\tau_{k+1}}{\tau_k}\right) - \left(\frac{\tau_{k+1}}{\tau_k}\right)^2\right] \\
\gamma_k &= -\frac{6}{\tau_k}\left[\left(\frac{\tau_{k+1}}{\tau_k}\right) - \left(\frac{\tau_{k+1}}{\tau_k}\right)^2\right] \\
\delta_k &= 3\left(\frac{\tau_{k+1}}{\tau_k}\right)^2 - 2\left(\frac{\tau_{k+1}}{\tau_k}\right)
\end{aligned}
\tag{8}
$$

with the time to go given by

$$\tau_k = T_F - T_k \quad , \quad \tau_{k+1} = T_F - T_{k+1} \tag{9}$$

We then transfer to the measurement variables with Eq. (4) in the form

$$y^\dagger = g(X^\dagger), \text{ (nonlinear)} \tag{10}$$

The best estimate of the state at T_{k+1} is then given by Kalman's relation

$$\hat{y}_{k+1} = y^\dagger_{k+1} + K_{k+1}\,(z_{k+1} - Hy^\dagger_{k+1}) \tag{11}$$

where z_{k+1} are the actual measurements, H is a rectangular matrix of ones and zeros that picks from the $y^\dagger_{k+1}$ vector those elements that correspond to the actual measurements, and K_{k+1} is the Kalman gain (yet to be calculated). We then transform back to the state variables with Eq. (5) in the form

$$\hat{X}_{k+1} = \ell(\hat{y}_{k+1}), \text{ (nonlinear)} \tag{12}$$

The Kalman gain is calculated by

$$K_{k+1} = M_{k+1/K}\, H^T(H\, M_{k+1/k}\, H^T+N)^{-1} \tag{13}$$

where N is a diagonal square matrix of the variances of the measurement errors and $M_{k+1/k}$ is the transferred covariance matrix of measurement variables calculated by

$$M_{k+1/k} = \psi_k\, M_{k/k}\, \psi_k^T \tag{14}$$

where, ψ_k is the state transition matrix for the measurement variables and $M_{k/k}$ is the measurement variables covariance matrix at T_k. (Note that no state disturbance has been included) It is here in the construction of ψ_k that approximations are made. Mahra observed that

$$\psi_k = \left(\frac{\partial y_{k+1}}{\partial y_k}\right)\left(\frac{\partial y_k}{\partial x_k}\right)\left(\frac{\partial x_k}{\partial y_{k+1}}\right)$$

or,

$$\psi_k = \left(\frac{\partial g}{\partial x}\right)_{k+1} \Phi_k \left(\frac{\partial \ell}{\partial y}\right)_k \tag{15}$$

The matrices $(\partial\ell/\partial y)$ and Φ_k are available from the best estimate of state at T_k. To form $(\partial g/\partial x)_{k+1}$, we use the first estimates at T_{k+1} obtained by linear extrapolation from T_k. The matrices $(\partial\ell/\partial y)$ and $(\partial g/\partial x)$ are not written out here. They can be found in Mahra's paper.[9] All that remains is to propagate the covariances to the next time and this is done with the usual relation

$$M_{k+1/k+1} = (I - K_{k+1}\, H)\, M_{k+1/k} \tag{16}$$

IV. Guidance and Navigation Evaluation Procedure

Performance evaluation of the overall guidance and navigation scheme requires accurate numerical simulation. A covariance analysis alone will not suffice because of the nonlinearities of the basic dynamics. Fig. 3 is a schematic of the procedure employed. Starting at a time T_k, an estimate of the state $\hat{x}_k$ is presumed available. For evaluation, the exact state x_k is also specified at this time. The estimate $\hat{x}_k$ is put into the guidance law

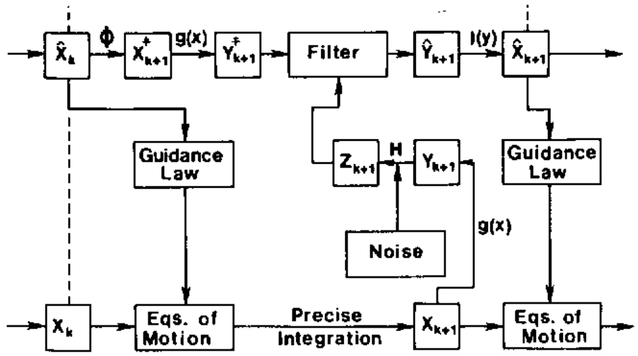

Fig. 3. Evaluation Scheme

to generate the SEP thrusting to be used. The equations of motion including the full effect of the sun and Keplerian motion of the target are then integrated accurately (fourth-order Runge-Kutta) to a time T_{k+1} when measurements will be available. The integrated state x_{k+1} is then transformed to measurement variables and appropriate noise added to simulate actual measurements z_{k+1}. To represent the onboard computations, the state $\hat{x}_k$ is propagated to time T_{k+1} through the transition matrix Φ_k. The nonlinear transformation $g(x)$ to measurement variables is then made to give a first estimate $y^{\dagger}$. The Kalman gain is then calculated with Eqs. (13), (14), (15). The filtered estimate $\hat{y}_{k+1}$ is then made with Eq. (11). This estimate is then transformed nonlinearly with $\ell(y)$ to obtain the new state estimate $\hat{x}_{k+1}$. And so on. The computer program for these computations has been given the acronym GANDER for Guidance and Navigation Development and Evaluation Routine.

V. Results

The computations for deterministic evaluation of the guidance algorithm were initiated at ranges as large as 2×10^6 km. However, such ranges are not possible for onboard range and range rate measurement with equipment that fits reasonable weight or power requirements. We could find no information that gives specific distance and accuracy limits for various weight and power allotments except at ranges less than 2000 km.[10] Several discussions led us to believe that 50,000 km. range and 10 measurements per day are conservative limits. At 50,000 km., optimal trajectory studies [6] and our deterministic investigations indicate relative velocity of 20,000 km/day is appropriate. The 50,000 km. range and 20,000 km/day relative velocity correspond to a point in time about 5 days before rendezvous.

Since specific accuracies could not be identified, we conducted evaluations with two assumed measurement error sets representing accurate and rough measurements. Angular measurements from onboard science TV gives accuracies of about 20 arc seconds.[1,2] To allow for onboard implementation, we chose twice this level of error for the angular measurements (.0002 radians or 41.3 arc sec). This and other error levels used are shown in Table 1.

Table 1. Measurement error sets

STANDARD DEVIATION	SET I	SET II
Angular Error	41.3 arc sec	41.3 arc sec
Range Error	.002 Range	.03 Range
Range Rate Error	4.63 cm/sec	1.15 m/sec

The remaining information necessary before evaluations can be made is the error in relative state information between spacecraft and target. It is obvious if final phase of terminal guidance is initiated at 50,000 km., that an ephemeris error of 100,000 km. (such as for Encke) cannot be tolerated. A preliminary study of onboard pre-final phase orbit determination indicates an improvement of relative position knowledge by as much as a factor of 20 by use of onboard relative angular measurements only. This result was obtained on the basis that the principal error in target ephemeris is time of periapsis passage. We assumed half of the estimated improvement and used 10,000 km. error in each position component. An error of 1000 km/day (11.5 m/sec) was assumed in each velocity component. The initial state covariance was constructed using these values as the standard deviations.

In initial computer runs, a gross filter divergence was found as expected. The divergence is obviously due to the modeling error introduced by approximations in the system dynamics. The covariance matrix rapidly decreased in size and new measurements were not weighted enough. One method of handling this problem is to introduce process noise directly into the differential equations. This has worked quite successfully before [1,2] but does introduce additional complexity in filter computations. A constant or adjustable matrix could also be added to the covariance matrix to control size of the principal elements. A nearly equivalent procedure was decided upon. After several guidance cycles, the covariance matrix had reduced in size considerably and we simply "froze" the matrix at this point. Results showed that a "freeze" after 10 guidance cycles or one day gave reasonable results. Certainly, this procedure is conservative. Of course, a procedure for better management should be developed for any actual system. But, our objective is a first evaluation and if reasonable results can be obtained with the crude freeze, then only improvement can be expected with further development.

The navigation errors for the two data sets are shown in Figs. 4 and 5. A standard computer routine was used to generate noise to simulate actual measurements. Different error sequences were used for each data type. Three different groups of the two runs for the accurate and rough data sets were investigated. There were no great differences in the results for the three groups. All groups were targeted at a point 100 km. in each coordinate from the target.

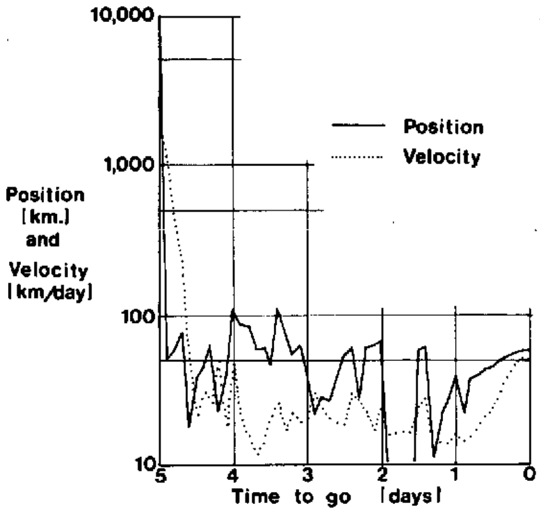

Fig. 4. Navigation Errors for Data Set I

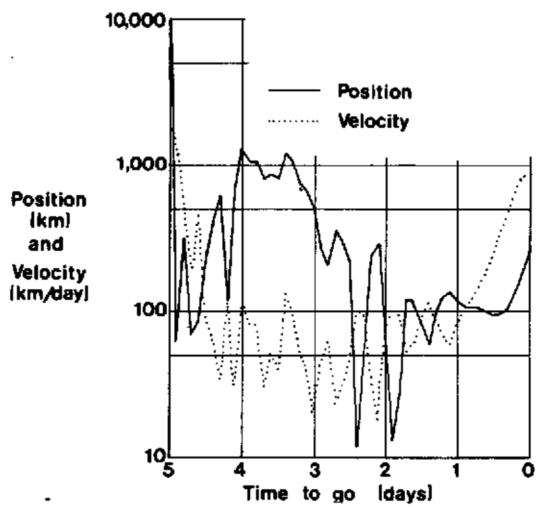

Fig. 5. Navigation Errors for Data Set II

For the accurate data Set I, the terminal rendezvous errors were about 58 km. and 50 km/day (0.58 m/sec). For the rough data Set II, the corresponding numbers were 280 km. and 860 km/day (10.0 m/sec). An examination of the error histories indicates, however, that much better results could be obtained with better statistical management. After initiation of the procedure at 5 days time to go, there is a rapid decrease in error from the initial values of 17,320 km. and 1732 km/day. It would seem that a better freeze time than 1 day could have been chosen. But, again, we were not interested in forcing the results. By a time of about one day before rendezvous, the errors settled down to about 15 km. and 25 km/day for the accurate Set I and to about 100 km. and 100 km/day for the rough Set II. After this time there is what appears to be a filter divergence to the final values. Examination of the differences in the extrapolated first state values and the filtered estimates confirm this is the case. No serious attempt was made to correct this divergence since the freezing procedure would probably not be used in any actual implementation.

The SEP thrust accelerations required for the two data sets are given in Fig. 6. The levels do not exceed values that are reasonable for proposed SEP systems. For the accurate Set I, the thrust is essentially constant over the whole period with deviations of only about the 5% error that may be expected from the SEP thrustors. For the rough SET II the deviations are larger but not extreme. The initial lower value arises because on the first few guidance cycles the vehicle does not know where it is. The increase during the last half day also

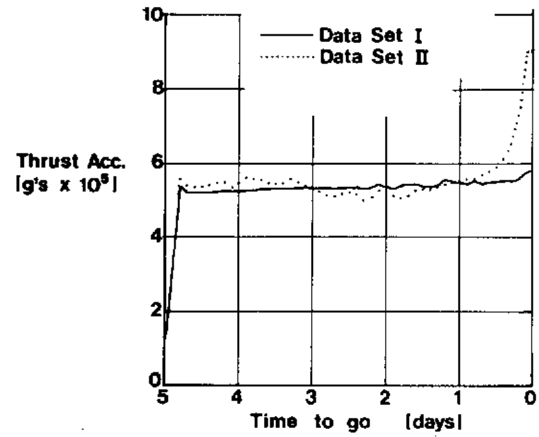

Fig. 6. Thrust Histories

arises because of navigation error. Examination of the detailed trajectories shows that the vehicle is on track to rendezvous. The filter problem near rendezvous discussed in the previous paragraph is the cause of thrust increase. In the sample data form in which the guidance equations are written, there is no numerical singularity except at rendezvous. In the case of the rough data Set II, fixing the thrust at a constant value of about 5.5×10^{-6} actually leads to more accurate rendezvous than following the incorrect filter estimates during the last half day, indicating the terminal point control problems may be helped by reducing frequency of guidance updates near the end of the trajectory so as to avoid small corrections in very small time. The nearly constant SEP level over most of the path also indicates a reduced guidance update frequency may be possible or desirable there as well.

VI. Conclusions

The filtering procedure used was certainly not the best that can be envisioned. Addition of process noise, or a basic improvement in the extrapolation of estimated state would give major improvement of results. Also, the study did not include errors in the SEP thrust level. But thrust level changes due to navigation errors were as large or larger than the 5% expected with the SEP thrustors. Of course, the SEP errors must be included in any more detailed analysis. We conclude that onboard navigation is possible without unreasonable accuracy requirements for onboard measuring equipment. But further investigation is clearly necessary to obtain definitive results.

For further study we suggest that the following be done:

1. Investigate methods of reducing modeling error with emphasis on ease of onboard implementation.

2. Include SEP thrust errors and a constraint for constant thrust level and direction rather than the linearly varying model now employed. Errors may be handled as noise or perhaps estimated as new state variables in the filtering process.

3. Investigate methods of thrust level control not only at the end point, but during the whole terminal phase. While excessive thrust was never encountered in our investigation, other missions may have to contend with gross off-nominal conditions that can call for excessive thrust unless an automatic control is incorporated in the procedure.

4. Investigate onboard methods of target orbit determination that will reduce the target ephemeris uncertainties.

5. Investigate the possibility of dispensing with some of the measurements used here with the objective of simplifying onboard systems.

6. Determine instrument capabilities, power requirements, weight, etc., for onboard measurements up to ranges of at least 50,000 km. and further, if possible.

References

1. Gardner, J. A., "Solar Electric Propulsion System Integration Technology (SEPSIT), Final Report," JPL Technical Memo 33-583, Vol. III, Nov. 1972.

2. Jacobson, R. A., "A Constrained Discrete Optimal Guidance Strategy for Low-Thrust Space Flight." Presented at AAS/AIAA Astrodynamics Conference, Vail, Colorado, July 16-18, 1973.

3. Cherry, G. W., "A Unified Explicit Technique Performing Orbital Insertion, Soft Landing, and Rendezvous with a Throttleable Rocket-Propelled Space Vehicle," MIT Instrumentation Laboratory Report R-417, MIT, Cambridge, August 1963.

4. Abercrombie, G. E., "Two Guidance Algorithms for Rendezvous Missions to Comets and Asteroids," Prepared under Contract NAS8-27664, Auburn University, December 1972.

5. Bryson, A. E., and Ho, Yu-Chi, Applied Optimal Control, Guin and Co., Waltham, Mass. (1969).

6. Friedlander, A. L., private communication.

7. Schlec, F. H., Standish, C. J., and Toda, N. F., "Divergence of the Kalman Filter," AIAA Journal, Vol. 5, No. 6, June 1967, pp. 1114-1120.

8. Jazwinski, A. H., Stochastic Processes and Filtering Theory, Academic Press, New York (1970), pp. 301-305.

9. Mehra, R. K., "A Comparison of Several Nonlinear Filters for Reentry Vehicle Tracking," IEEE Trans. on Auto. Control, Vol. AC-16, No. 4, August 1971, pp. 307-319.

10. Hung, J. C., "Autonomous Target Relative Navigation," Systems Research Group, Dept. of Electrical Engineering, University of Tennessee, Sci. Report S-24, January 31, 1973.

APPENDIX - COMPUTER PROGRAM GANDER

General Program Description

The GANDER Program is written in FORTRAN IV and used with the IBM 370/155. The program is a research tool, not a developed production routine. The steps in the simulation and the names of the subroutines that carry out these steps are as follows.

A fourth order Runge-Kutta subroutine, RUNKUT, is used to integrate the dynamic forces in GOFZ$ over subintervals of length DT, at each time step (DELT), observations are taken in OBSERV and the filter is used to predict the state in FILTER. The information generated is then put out on Unit 6 and terminal conditions are checked in CYCLOT. Program sequencing and execution is controlled by subroutine CYCLE. Subroutine TARGET is used to generate the comet's position. The name, NOISE, is a dummy name for the functions URAND (Uniformly distributed random numbers) and GRAND (Gaussian distributed random numbers). Ten (10) independent noise channels are shared by these two functions.

Subroutine Names and Descriptions

MAIN • Reads in system data and calls CYCLE.

CYCLE • Controls sequence of operation and transfer of data between XT and SP.

RUNKUT • Fourth Order Runge-Kutta Integrator. Dynamics are provided by DOFX$ and guidance by GOFX$. Called by CYCLE. Performs N integrations of step size DT at each call.

• Entries:
RKINIT called by CYCLE.
Initialize internal variables and read in XT.

DOFX$ • Compute contribution of dynamics to $\dot{X}$. J is the index of the components of XT.

Entries:

DOFX$: Compute data to be used by all components.

DOFX: Compute each component.

DXINIT: Initialize internal constants and read in comet data.

SPECIAL - Calls target for comet position.

GOFX$ • Same as DOFX$ except XP is used as variable. May call GRAND.

FILTER • User supplied algorithm to calculate XP. (Basic Kalman filter used in present listing)

• Entries:
FLINIT: Used to initialize arrays.

SPECIAL: Variable ICNT is used to bypass covariance update section after preselected number of cycles. Calls MINV

OBSERV • Generates ZT; may call GRAND.

CYCLOT • Outputs data and checks for end of run.

• Entries:
CYCLOT: Output data

TERMIN: Check for end conditions satisfied.

RECAP: If end conditions satisfied; output minimum normed distance, velocity and associated times.

CYINIT: Initialize internal constants.

TARGET • Comet's position by solution of Kepler's Equation.

MINV • Gaussian elimination inversion routine.

SPECIAL writes square (nxn) matrix as a vector of length n^2. (Modified IBM SSP)

GRAND • Generates Gaussian distributed random noise with given mean (RMEAN) and standard deviation (STDDEV).

LSCNT • Noise channel number. Calls URAND.

URAND • Generates random numbers over the interval [0,1]. (Modified IBM SSP).

BLOCK DATA initializes seed numbers for URAND

Variable Names and Definitions

XT	True state vector.
XP	Predicted state vector (loaded in GXINIT)
XE	XT-XP error in state
ZT	True observations
ZP	Predicted observations
ZE	ZT-ZP error in observations (RESIDUALS)
L$L	One BYTE logical array used to control sequencing of simulator (partially implemented).
L$E	One BYTE logical array for use by error monitor (Experimantal program control, not implemented).
DT	Integration stepsize (true position).
DELT	Guidance update stepsize (predicted position) must be an INTEGRAL multiple of DT.

N	DELT/DT (an integer (exactly))
TI	Integration start time
IF	Integration end time
ISLEN	Number of elements in the state vector (XT and XP).
IOLEN	Number of elements in the observation vector (ZT and ZP).
C	Rendezvous stand-off distance.
A	Target semi-major axis
RN	Target mean motion.
EPS	Target eccentricity.
EO	Target eccentric anomaly.
TSTAR	Guidance initiation time.

Input Data

CARD NO.	CONTENTS	FORMAT	READ BY
	REQUIRED SYSTEM DATA		
1	Title Card (L$L)	5A8	Main
2	Run Time Logical Flags (L$E)	80L1	Main
3	Run Time Error Flags	80L1	Main
4	N, ISLEN, IOLEN	8I3	Main
5	TI, TF, DT, DELT	8F10.0	Main
	USER DATA		
6	XT (initial conditions)	8F10.0	RKINIT
7	A,RN,EPS,EO,TSTAR	8F10.0	DXINIT
8	C	8F10.0	DXINIT

Subroutine Initialization Entries

SUBROUTINES AND LINE NUMBERS	ENTRY
GXINIT(J)	
34, 35, 36	Initial position error, kilometers.
37, 38, 39	Initial velocity error, km/day.
FLINIT	
188 Thru 193	Initial covariance matrix (diagonal elements only in present listing).
OVINIT	
61	Standard deviation of angular measurement error (EANG)
62, 63	Fix logical if statements. Do not change (RMIN and RDMIN).
OBSERV	
16, 17, 28, 34, 40, 46	Mean, St'd. Dev., noise channel (To generate rendom numbers form GRAND). Ten noise channels available.
27, 45	Range error Std. dev. RFRAF(1)
33, 39	Range rate error, std. dev. RFRAF RFRAF(4).
FILTER	
141	Number of guidance steps before covariance freeze (10 steps in present listing). Remove card for no freeze.

GANDER Program Listing

FORTRAN IV G LEVEL 21 MAIN

```
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/OFFSET/C(3)
      DIMENSION B(4,6),RN(4,4),T6(4,4)
      DATA LF,LT/F,T/
  502 FORMAT(80L1)
  503 FORMAT(5A8)
  504 FORMAT(8I3)
  505 FORMAT(8F10.0)
  602 FORMAT(1H0,80L1)
  603 FORMAT(1H0,5A8)
  604 FORMAT(1H0,'   INPUT CARD LIST')
  606 FORMAT(1H ,////)
  607 FORMAT(1H0,8I5)
  608 FORMAT(1H0,1P6D12.5)
      WRITE(6,604)
      READ(5,503)TITLE
      WRITE(6,603)TITLE
      READ(5,502)L$L
      WRITE(6,602)L$L
      READ(5,502)L$E
      WRITE(6,602)L$E
      READ(5,504)N,ISLEN,IOLEN
      WRITE(6,607)N,ISLEN,IOLEN
      READ(5,505)TI,TF,DT,DELT
      WRITE(6,608)TI,TF,DT,DELT
      WRITE(6,606)
      CALL CYCLE
      STOP
      END
```

FORTRAN IV G LEVEL 21 CYCLE

```
      SUBROUTINE CYCLE
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/OFFSET/C(3)
      DATA LF,LT/F,T/
      DIMENSION B(4,6),RN(4,4),T6(4,4)
C     **** INITIALIZE SUBROUTINES ****
      CALL RKINIT
      DUM=DXINIT(1)
      DUM=GXINIT(1)
      DUM=UHINIT(1)
      CALL CYINIT
      CALL FLINIT
      CALL OVINIT
C     **** COMPUTE TRAJECTORIES ****
    1 CALL PUNKUT
      IF(L$L(2))GO TO 2
```

```
      CALL OBSERV
      CALL FILTER(8,RN,T6)
      GO TO 3
    2 DO 4 I=1,ISLEN
    4 XP(I)=XT(I)
C    **** OUTPUT CYCLE DATA ****
    3 CALL CYCLOT
C    **** MONITOR SECTION ****
      CALL TERMIN
      IF(L$L(2))GO TO 5
      IF(L$L(3))GO TO 5
      IF(L$L(6))GO TO 1
      GO TO 7
    5 DO 6 I=1,ISLEN
    6 XP(I)=XT(I)
      IF(L$L(6))GO TO 1
C    **** OUTPUT SECTION ****
    7 CALL RECAP
      RETURN
      END
```

```
FORTRAN IV G LEVEL  21                    TARGET

      SUBROUTINE TARGET(S,T)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION S(3)
      COMMON/VAR1/A,RN,EPS,ET,TSTAR,DET
  601 FORMAT(1H0,'CONVERGENCE= ',1PD12.5,' KEPEQ')
      EO=ET
      ET=ET+DET
      DO 1 I=1,100
      SINET=DSIN(ET)
      COSET=DCOS(ET)
      F=RN*(T+TSTAR)-ET+EPS*SINET
      DF=EPS*COSET-1.0D0
      ET=ET-F/DF
      DIF=DABS(F/DF)
      IF(DIF.LT.1.0D-10)GO TO 2
    1 CONTINUE
      WRITE(6,601)DIF
    2 DET=ET-EO
      S(1)=A*(COSET   -EPS)
      S(2)=A*DSQRT(1.0D0-EPS*EPS)*SINET
      S(3)=0.0D0
      RETURN
      END
```

```
      SUBROUTINE RUNKUT
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      LOGICAL*1 LS1,LS2
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/OFFSET/C(3)
      DIMENSION XINT(6),SUM(6)
      DATA LF,LT/F,T/
      L$L(12)=LT
      DO 1 ICYCLE=1,N
      DO 33 I=1,ISLEN
   33 SUM(I)=0.0D0
      L$L(10)=LT
      DO 10 I1=1,4
      LS1=I1.EQ.2. OR.I1.EQ.3
      LS2=I1.EQ.4
      L$L(11)=I1.EQ.3
      F=F1
      FS=F5
      IF(LS1)F=F2
      IF(LS1)FS=F3
      IF(LS2)FS=F4
      TS=TIME+DT*FS
      DO 20 I=1,ISLEN
20    XINT(I)=XT(I)+FS*XINT(I)
      DO 31 I=1,NP1
      J=I-1
      IF(J.GT.0) GO TO 2
      DUM=DOFX$(J,TS)
      DUM=GOFX$(J,TS)
      L$L(12)=LF
      GO TO 31
2     XINT(J)=DT*(DOFX(J)+GOFX(J))
      SUM(J)=SUM(J)+F*XINT(J)
31    CONTINUE
      L$L(10)=LF
10    CONTINUE
      TIME=TIME+DT
      DO 11 I=1,ISLEN
11    XT(I)=XT(I)+SUM(I)
    1 CONTINUE
      RETURN
      ENTRY RKINIT
      READ(5,501)(XT(I),I=1,ISLEN)
      WRITE(6,601)(XT(I),I=1,ISLEN)
  501 FORMAT(8F10.0)
  601 FORMAT(1H0,1P8D12.5)
      F1=1.0D0/6.0D0
      F2=2.0D0*F1
      F3=1.0D0/2.0D0
      F4=1.0D0
      F5=0.0D0
      TIME=TI
      NP1=ISLEN+1
```



```
      DO 32 I=1,ISLEN
   32 XINT(I)=XT(I)
      RETURN
      END
```

```
      FUNCTION DOFX$(J,TS)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SYSTEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/VAR1/A,RN,EPS,EO,TSTAR,DET
      COMMON/OFFSET/C(3)
      DIMENSION D(3),S(3)
      DATA LF,LT/F,T/
99601 FORMAT(1H0,I4,'IMPROPER INDEX *DOFX')
      IF(.NOT.L$L(11))CALL TARGET(S,TS)
      S2=0.0D0
      D2=0.0D0
      DO 1 I=1,3
      D(I)=S(I)+C(I)+XT(I)
      D2=D2+D(I)**2
    1 S2=S2+S(I)**2
      DN=DSQRT(D2)
      SN=DSQRT(S2)
      RAT1=GM/(SN*SN*SN)
      RAT2=(SN/DN)**3
      DOFX$=0.0D0
      IF(TF-TIME.GT.DELT)RETURN
      IF(.NOT.L$L(10))RETURN
      WRITE(6,602)TIME,XT
  602 FORMAT(1H0,'END STATE ',2X,'TIME=',F10.3/1H ,1P6D12.5)
      RETURN
      ENTRY DOFX(J)
      GO TO (99999,99999,99999,99998,99998,99998),J
      WRITE(6,99601)J
      DOFX=0.0D0
      L$E(2)=LT
      RETURN
99999 DOFX=XT(J+3)
      RETURN
99998 DOFX=RAT1*(S(J-3)-D(J-3)*RAT2)
      RETURN
      ENTRY DXINIT(J)
      READ(5,501)A,RN,EPS,EO,TSTAR
      EO=RN*TSTAR
      WRITE(6,601)A,RN,EPS,EO,TSTAR
  501 FORMAT(8F10.0)
  601 FORMAT(1H0,1P8D12.5)
      READ(5,501)C
      WRITE(6,601)C
      GM=9.9054902D0
      DET=1.0D-3
      DXINIT=0.0D0
      RETURN
      END
```

```
      FUNCTION GOFX$(J,TS)
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/FORCE/F(3)
      COMMON/OFFSET/C(3)
      DATA LF,LT/F,T/
99601 FORMAT(1HO,I4,'IMPROPER INDEX *GOFX')
      IF(L$L(12))TIM1=TS
      TAUO=TF-TIM1
      TAU=TF-TS
      TRAT1=(6.0D0/TAUO**2)*(1.0D0-2.0D0*(TAU/TAUO))
      TRAT2=(2.0D0/TAUO)*(1.0D0-3.0D0*(TAU/TAUO))
      GOFX$=0.0D0
      RETURN
      ENTRY GOFX(J)
      GO TO (99999,99999,99999,99998,99998,99998),J
      WRITE(6,99601)J
      L$E(3)=LT
      GOFX=0.0D0
      RETURN
99999 GOFX=0.0D0
      RETURN
99998 F(J-3)=TRAT1*XP(J-3)+TRAT2*XP(J)
      GOFX=F(J-3)
      RETURN
      ENTRY GXINIT(J)
      DO 90001 I=1,ISLEN
90001 XP(I)=XT(I)
      XP(1)=XP(1)+10000.
      XP(2)=XP(2)+10000.
      XP(3)=XP(3)+10000.
      XP(4)=XP(4)+1000.
      XP(5)=XP(5)+1000.
      XP(6)=XP(6)+1000.
      GXINIT=0.0D0
      RETURN
      END
```

```
0001              SUBROUTINE FILTER(B,RN,T6)
0002              IMPLICIT REAL*8(A-H,O-Z)
0003              LOGICAL*1 L$L,L$E,LMON,LF,LT
0004              COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
0005              COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
0006              COMMON/SY$TEM/L$L(40),L$E(10)
0007              COMMON/M$NITR/LMON(20)
0008              COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
0009              COMMON/KALMAN/RM(6,6),FILT(6,4),U(6)
0010              COMMON/OFFSET/C(3)
0011              DIMENSION T1(6),YL(6),Y(6),T2(6,6),T3(6,6),T4(6,6),QP(6,6),
                 1B(IOLEN,6),RN(IOLEN,IOLEN),T5(6,6),T6(IOLEN,IOLEN),PHI(6,6)
0012              DIMENSION LL(6),MM(6)
0013              DATA LF,LT/F,T/
0014              WRITE(6,80)IOLEN
0015           80 FORMAT(20X,'IOLEN=',I2)
0016              ICNT=ICNT+1
0017              DO 38 I=1,IOLEN
0018              DO 32 J=1,IOLEN
0019           32 RN(I,J)=0.0D0
0020              DO 38 J=1,ISLEN
0021           38 B(I,J)=0.0D0
0022              IF(IOLEN.EQ.2)GO TO 34
0023              IF(L$L(20).AND.IOLEN.EQ.3)GO TO 37
0024              DO 33 I=1,3
0025           33 B(I,I)=1.0D0
0026              IF(IOLEN.EQ.4)B(4,6)=1.0D0
0027              GO TO 36
0028           37 B(1,2)=1.0D0
0029              B(2,3)=1.0D0
0030              B(3,6)=1.0D0
0031              DO 39 I=1,3
0032           39 RN(I,I)=RFRAF(I+1)**2
0033              GO TO 35
0034           34 B(1,2)=1.0D0
0035              B(2,3)=1.0D0
0036              RN(1,1)=RFRAF(2)**2
0037              RN(2,2)=RFRAF(3)**2
0038              GO TO 35
0039           36 DO 30 I=1,IOLEN
0040           30 RN(I,I)=RFRAF(I)**2
0041           35 TAUK1=TF-TIME
0042              TAUK=TAUK1+DELT
0043              RAT=TAUK1/TAUK
0044              RAT2=RAT*RAT
0045              RAT3=RAT2*RAT
0046              A=3.0D0*RAT2-2.0D0*RAT3
0047              DIF=RAT-RAT2
0048              F=TAUK1*DIF
0049              D=3.0D0*RAT2-2.0D0*RAT
0050              E=-6.0D0*DIF/TAUK
0051              DO 31 I=1,3
0052              PHI(I,I)=A
0053              PHI(I,I+3)=F
0054              PHI(I+3,I)=E
0055              PHI(I+3,I+3)=D
0056           31 CONTINUE
            C     PREDICT X STATE
```

```
      DO 10 I=1,ISLEN
      T1(I)=0.0D0
      DO 10 J=1,ISLEN
   10 T1(I)=T1(I)+PHI(I,J)*XP(J)
C     TRANSFORM TO Y SYSTEM
      Y(1)=DSQRT((T1(1)+C(1))**2+(T1(2)+C(2))**2+(T1(3)+C(3))**2)
      Y(2)=(T1(1)+C(1))/Y(1)
      Y(3)=(T1(2)+C(2))/Y(1)
      W=(T1(3)+C(3))/Y(1)
      Y(4)=((T1(1)+C(1))*T1(4)+(T1(2)+C(2))*T1(5)+(T1(3)+C(3))*T1(6))
     */Y(1)
      Y(5)=(T1(4)-Y(4)*Y(2))/Y(1)
      Y(6)=(T1(5)-Y(4)*Y(3))/Y(1)
      WD=(T1(6)-Y(4)*W)/Y(1)
C     COMPUTE LEFT PARTIAL DERIVATIVE
      T2(1,1)=Y(2)
      T2(1,2)=Y(3)
      T2(1,3)=W
      T2(2,1)=(1.0D0-Y(2)**2)/Y(1)
      T2(2,2)=-Y(2)*Y(3)/Y(1)
      T2(2,3)=-Y(2)*W/Y(1)
      T2(3,1)=T2(2,2)
      T2(3,2)=(1.0D0-Y(3)**2)/Y(1)
      T2(3,3)=-(Y(3)*W)/Y(1)
      DO 11 I=1,3
      DO 11 J=1,3
   11 T2(I+3,J+3)=T2(I,J)
      T2(4,1)=Y(5)
      T2(4,2)=Y(6)
      T2(4,3)=WD
      T2(5,1)=-(2.0D0*Y(2)*Y(5)+(1.0D0-Y(2)**2)*Y(4)/Y(1))/Y(1)
      T2(5,2)=-(Y(3)*Y(5)+Y(2)*Y(6)-Y(2)*Y(3)*Y(4)/Y(1))/Y(1)
      T2(5,3)=-(W*Y(5)+Y(2)*WD-Y(2)*W*Y(4)/Y(1))/Y(1)
      T2(6,1)=T2(5,2)
      T2(6,2)=-(2.0D0*Y(3)*Y(6)+(1.0D0-Y(3)**2)*Y(4)/Y(1))/Y(1)
      T2(6,3)=-(W*Y(6)+Y(3)*WD-Y(3)*W*Y(4)/Y(1))/Y(1)
C     COMPUTE RIGHT PARTIAL DERIVATIVE
      T3(1,1)=YL(2)
      T3(4,4)=YL(2)
      T3(1,2)=YL(1)
      T3(4,5)=YL(1)
      T3(2,1)=YL(3)
      T3(5,4)=YL(3)
      T3(2,3)=YL(1)
      T3(5,6)=YL(1)
      T3(3,1)=WL
      T3(6,4)=WL
      T3(3,2)=-YL(2)*YL(1)/WL
      T3(6,5)=T3(3,2)
      T3(3,3)=-YL(3)*YL(1)/WL
      T3(6,6)=T3(3,3)
      T3(6,1)=WDL
      T3(6,2)=-(XP(4)/WL)+YL(1)*YL(2)*WDL/WL**2
      T3(6,3)=-(XP(5)/WL)+YL(1)*YL(3)*WDL/WL**2
C     COMPUTE PSI
      DO 12 I=1,ISLEN
      DO 12 J=1,ISLEN
      T4(I,J)=0.0D0
```

```
            DO 12 K=1,ISLEN
            DO 12 L=1,ISLEN
         12 T4(I,J)=T4(I,J)+T2(I,K)*PHI(K,L)*T3(L,J)
C        PREDICT COVARIANCE
            DO 13 I=1,ISLEN
            DO 13 J=1,ISLEN
            T5(I,J)=QP(I,J)
            DO 13 K=1,ISLEN
            DO 13 L=1,ISLEN
         13 T5(I,J)=T5(I,J)+T4(I,K)*RM(K,L)*T4(J,L)
C        COMPUTE THE FILTER
            DO 14 I=1,IOLEN
            DO 14 J=1,IOLEN
            T6(I,J)=RN(I,J)
            DO 14 K=1,ISLEN
            DO 14 L=1,ISLEN
         14 T6(I,J)=T6(I,J)+B(I,K)*T5(K,L)*B(J,L)
            CALL MINV(T6,IOLEN,16,LL,MM,D)
            DO 15 I=1,ISLEN
            DO 15 J=1,IOLEN
            FILT(I,J)=0.0D0
            DO 15 K=1,ISLEN
            DO 15 L=1,IOLEN
         15 FILT(I,J)=FILT(I,J)+T5(I,K)*B(L,K)*T6(L,J)
C        COMPUTE PREDICTED OBSERVATIONS
            DO 16 I=1,IOLEN
            ZP(I)=0.0D0
            DO 16 J=1,ISLEN
         16 ZP(I)=ZP(I)+B(I,J)*Y(J)
C        COMPUTE THE ERROR IN OBSERVATIONS
            DO 17 I=1,IOLEN
         17 ZE(I)=ZT(I)-ZP(I)
C        UPDATE Y
            DO 18 I=1,ISLEN
            T1(I)=Y(I)
            DO 18 J=1,IOLEN
         18 T1(I)=T1(I)+FILT(I,J)*ZE(J)
C        UPDATE COVARIANCE
            IF(ICNT.GT.10.AND.ICNT.LT.32)GO TO 99
            IF(ICNT.GT.40) GO TO 99
            DO 19 I=1,ISLEN
            DO 19 J=1,ISLEN
            T4(I,J)=0.0D0
            DO 19 K=1,IOLEN
         19 T4(I,J)=T4(I,J)+FILT(I,K)*B(K,J)
            DO 20 I=1,ISLEN
            DO 20 J=1,ISLEN
            T4(I,J)=-T4(I,J)
            IF(I.EQ.J)T4(I,J)=T4(I,J)+1.0D0
         20 CONTINUE
            DO 21 I=1,ISLEN
            DO 21 J=1,ISLEN
            RM(I,J)=0.0D0
            DO 21 K=1,ISLEN
         21 RM(I,J)=RM(I,J)+T4(I,K)*T5(K,J)
         99 CONTINUE
C        SAVE Y(K+1,K+1)
            DO 22 I=1,ISLEN
```

```
0161          22 YL(I)=T1(I)
            C    CONVERT TO X
0162             XP(1)=YL(2)*YL(1)-C(1)
0163             XP(2)=YL(3)*YL(1)-C(2)
0164             XP(4)=YL(1)*YL(5)+YL(4)*YL(2)
0165             XP(5)=YL(1)*YL(6)+YL(4)*YL(3)
0166             XP(3)=DSQRT(YL(1)**2-(XP(1)+C(1))**2-(XP(2)+C(2))**2)-C(3)
0167             XP(6)=(YL(1)*YL(4)-(XP(1)+C(1))*XP(4)-(XP(2)+C(2))*XP(5))/(XP(3)+
                6C(3))
0168             WL=(XP(3)+C(3))/YL(1)
0169             WDL=(XP(6)-YL(4)*WL)/YL(1)
0170             WRITE(6,912)(XP(J),J=1,6)
0171         912 FORMAT(6D12.5//)
0172             RETURN
0173             ENTRY FLINIT
            C    ZERO ARRAYS
0174             DO 101 I=1,ISLEN
0175             DO 102 J=1,ISLEN
0176             T2(I,J)=0.0D0
0177             T3(I,J)=0.0D0
0178             RM(I,J)=0.0D0
0179             PHI(I,J)=0.0D0
0180         102 QP(I,J)=0.0D0
0181             DO 101 J=1,IOLEN
0182         101 FILT(I,J)=0.0D0
            C    INITIALIZE VARIABLES
0183             YL(1)=DSQRT((XP(1)+C(1))**2+(XP(2)+C(2))**2+(XP(3)+C(3))**2)
0184             YL(2)=(XP(1)+C(1))/YL(1)
0185             YL(3)=(XP(2)+C(2))/YL(1)
0186             WL=(XP(3)+C(3))/YL(1)
0187             YL(4)=((XP(1)+C(1))*XP(4)+(XP(2)+C(2))*XP(5)+(XP(3)+C(3))*XP(6))
                */YL(1)
0188             YL(5)=(XP(4)-YL(4)*YL(2))/YL(1)
0189             YL(6)=(XP(5)-YL(4)*YL(3))/YL(1)
0190             RM(1,1)=1.0D8
0191             RM(2,2)=4.0D-02*(1-YL(2)**2)
0192             RM(3,3)=4.0D-02*(1-YL(3)**2)
0193             RM(4,4)=1.0D6
0194             RM(5,5)=1.0D-05*YL(2)**2
0195             RM(6,6)=1.0D-05*YL(3)**2
0196             WDL=(XP(6)-YL(4)*WL)/YL(1)
0197             ICNT=0
0198             RETURN
0199             END
```

```
      SUBROUTINE OBSERV
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/OFFSET/C(3)
      DATA LF,LT/F,T/
C   OBSERVATIONS
C     RFRAF(1 TO 4) = STDDEV FOR ZT(1 TO 4) RESPECTIVELY
      RANGE=DSQRT((XT(1)+C(1))**2+(XT(2)+C(2))**2+(XT(3)+C(3))**2)
      ZT(2)=(XT(1)+C(1))/RANGE
      ZT(3)=(XT(2)+C(2))/RANGE
      UAC=DARCOS(ZT(2))
      VAC=DARCOS(ZT(3))
      DG(2)=GRAND(0.0D0,EANG,2)
      DG(3)=GRAND(0.0D0,EANG,3)
      UAC=UAC+DG(2)
      VAC=VAC+DG(3)
      ZT(2)=DCOS(UAC)
      ZT(3)=DCOS(VAC)
      RFRAF(2)=DSQRT(EANG**2*(1.0D0-ZT(2)**2))
      RFRAF(3)=DSQRT(EANG**2*(1.0D0-ZT(3)**2))
      IF(L$L(20)) GO TO 10
      IF(RANGE.GT.RMIN)GO TO 2
      ZT(1)=RANGE
      RFRAF(1)=DABS(ZT(1)*3.0D-02)
      DG(1)=GRAND(0.0D0,RFRAF(1),1)
      ZT(1)=ZT(1)+DG(1)
      IF(RANGE.LT.RMIN.AND.RANGE.GT.RDMIN)GO TO 3
      IOLEN=4
      ZT(4)=((XT(1)+C(1))*XT(4)+(XT(2)+C(2))*XT(5)+(XT(3)+C(3))*XT(6))
     */ZT(1)
      RFRAF(4)=1.0D02
      DG(4)=GRAND(0.0D0,RFRAF(4),4)
      ZT(4)=ZT(4)+DG(4)
      GO TO 5
   10 IF(RANGE.GT.RDMIN) GO TO 2
      ZT(4)=((XT(1)+C(1))*XT(4)+(XT(2)+C(2))*XT(5)+(XT(3)+C(3))*XT(6))
     */RANGE
      RFRAF(4)=1.0D02
      DG(4)=GRAND(0.0D0,RFRAF(4),4)
      ZT(4)=ZT(4)+DG(4)
      IF(RANGE.LT.RDMIN.AND.RANGE.GT.RMIN) GO TO 6
      IOLEN=4
      ZT(1)=RANGE
      RFRAF(1)=DABS(ZT(1)*3.0D-02)
      DG(1)=GRAND(0.0D0,RFRAF(1),1)
      ZT(1)=ZT(1)+DG(1)
      GO TO 5
    2 WRITE(6,1)(DG(LSCNT),LSCNT=2,3)
      IOLEN=2
      GO TO 4
    3 IOLEN=3
    5 WRITE(6,1)(DG(LSCNT),LSCNT=1,IOLEN)
    1 FORMAT(///,10X,'NOISE',5X,1P4D12.5)
```

```
FORTRAN IV G LEVEL  21                    OBSERV

          GO TO 4
        6 WRITE(6,1)(DG(LSCNT),LSCNT=2,4)
          IOLEN=3
        4 RETURN
          ENTRY OVINIT
          A=0.0D0
          EAVG=.0002
          RMIN=1.0D20
          RDMIN=1.0D20
          RETURN
          END
```

```
FORTRAN IV G LEVEL  21                    GRAND

          FUNCTION GRAND(RMEAN,STDDEV,ISLCT)
          IMPLICIT REAL*8(A-H,O-Z)
C      PURPOSE
C         COMPUTES A NORMALLY DISTRIBUTED RANDOM NUMBER WITH A GIVEN
C         MEAN AND STANDARD DEVIATION
          A=0.0D0
          DO 50 I=1,12
       50 A=A+URAND(ISLCT)
          GRAND=(A-6.0D0)*STDDEV+RMEAN
          RETURN
          END
```

```
FORTRAN IV G LEVEL  21                    URAND

          FUNCTION URAND(ISLCT)
          IMPLICIT REAL*8(A-H,O-Z)
          COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
          IY=IRAN(ISLCT)*65539
          IF(IY)5,6,6
        5 IY=IY+2147483647+1
        6 URAND=DFLOAT(IY)*4.656613D-10
          IRAN(ISLCT)=IY
          RETURN
          ENTRY URINIT(ISLCT)
          URAND=0.0D0
          URINIT=0.0D0
          RETURN
          END
```

```
FORTRAN IV G LEVEL  21                    BLK DATA

          BLOCK DATA
          IMPLICIT REAL*8 (A-H,O-Z)
          COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
          DATA IRAN/69800661,54218059,51070625,15239339,75892237,
         *10418327,81767867,59847821,52031357,26256073/
          END
```

```
      SUBROUTINE CYCLOT
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL*1 L$L,L$E,LMON,LF,LT
      COMMON/V$RBLE/XT(6),XP(6),XE(6),ZT(6),ZP(6),ZE(6)
      COMMON/T$MER/DELT,DT,TIME,TI,TF,N,ISLEN,IOLEN
      COMMON/SY$TEM/L$L(40),L$E(10)
      COMMON/M$NITR/LMON(20)
      COMMON/NOI$E/IRAN(10),DG(10),RFRAF(6)
      COMMON/KALMAN/ P(6,6),FILT(6,4),U(6)
      COMMON/FORCE/F(3)
      COMMON/OFFSET/C(3)
      DATA LF,LT/F,T/
  601 FORMAT(1H ,3X,'TRUE STATE VECTOR'/4X,1P6D12.5)
  603 FORMAT(1H ,'TIME=',1PD12.5)
  604 FORMAT(1H ,'NORMED DISTANCE=',1PD12.5,3X,'NORMED VELOCITY=',
     11PD12.5,3X,'NORMED FORCE=',1PD12.5)
  605 FORMAT(1H0,'$$$  RENDEZVOUS $$$')
  606 FORMAT(1H0,'MINIMUM NORMED DISTANCE=',1PD12.5,3X,'AT TIME=',
     1 1PD12.5)
  607 FORMAT(1H0,'MINIMUM NORMED VELOCITY =',1PD12.5,3X,'AT TIME=',
     1 1PD12.5)
  608 FORMAT(////)
  609 FORMAT(1H0,'**** OUT OF TIME ***')
  610 FORMAT(1H0,'FORCE VECTOR',/,1H ,1P3D12.5)
  611 FORMAT(1H0,3X,'PREDICTED STATE VECTOR'/4X,1P6D12.5)
  612 FORMAT(1H0,3X,'ERROR IN STATE VECTOR'/4X,1P6D12.5)
  613 FORMAT(1H0,3X,'TRUE OBSERVATIONS'/4X,1P6D12.5)
  614 FORMAT(1H0,3X,'PREDICTED OBSERVATIONS'/4X,1P6D12.5)
  615 FORMAT(1H0,3X,'RESIDUAL ERROR'/4X,1P6D12.5)
  617 FORMAT(1H0,3X,'COVARIANCE MATRIX')
  618 FORMAT(1H ,6X,1P6D12.5)
  619 FORMAT(1H1,10X,' SIMULATION RESULTS',////)
      IF(L$L(1))WRITE(6,619)
  620 FORMAT(1H ,3X,'NORMED POSITION ERROR = ',1P1D12.5,5X,'NORMED VELOC
     $ITY ERROR = ',1P1D12.5)
      FT=0.0D0
      T1=0.0D0
      T2=0.0D0
      DO 1 I=1,3
      F(I)=F(I)*CF1
      FT=FT+F(I)**2
      T1=T1+XT(I)*XT(I)
    1 T2=T2+XT(I+3)*XT(I+3)
      FT=DSQRT(FT)
      XS=DSQRT(T1)
      XV=DSQRT(T2)
      IF(L$L(9))GO TO 2
      IF(XS.GT.XO)GO TO 3
      XO=XS
      TXS=TIME
    3 IF(XV.GT.XVO)GO TO 2
      XVO=XV
      TXV=TIME
    2 WRITE(6,603)TIME
      WRITE(6,604)XS,XV,FT
      CHK=DABS(TF-TIME)
      IF(CHK.LT.DT)TF=TF+DELT
      IF(L$L(2))RETURN
```

```
      DO 99901 I$=1,ISLEN
99901 XE(I$)=XT(I$)-XP(I$)
      PNORM=DSQRT(XE(1)**2+XE(2)**2+XE(3)**2)
      VNORM=DSQRT(XE(4)**2+XE(5)**2+XE(6)**2)
      WRITE(6,610)F
      WRITE(6,601)XT
      WRITE(6,611)XP
      WRITE(6,612)XE
      WRITE(6,620)PNORM,VNORM
      WRITE(6,613)ZT
      WRITE(6,614)ZP
      WRITE(6,615)ZE
      WRITE(6,617)
      DO 6 I=1,ISLEN
    6 WRITE(6,618)(P(I,J),J=1,ISLEN)
      WRITE(6,608)
99902 L$L(1)=LF
      RETURN
      ENTRY TERMIN
      IF(XS.LT.1.0D-2.AND.XV.LT.1.0D-4)GO TO 4
      IF(TIME.GE.TF)GO TO 5
      RETURN
    4 WRITE(6,605)
      L$L(6)=LF
      RETURN
    5 WRITE(6,609)
      L$L(6)=LF
      RETURN
      ENTRY RECAP
      WRITE(6,606)XO,TXS
      WRITE(6,607)XVO,TXV
      RETURN
      ENTRY CYINIT
      XO=1.0D40
      XVO=1.0D40
      CF1=1.0D0/(9.80665D-3*8.64D4**2)
      RETURN
      END
```

```
              SUBROUTINE MINV(A,N,NSQ,L,M,BIGA)
              IMPLICIT REAL*8 (A-H,O-Z)
              DIMENSION A(NSQ),L(6),M(6)
C
C                DESCRIPTION OF PARAMETERS
C                    A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C                        RESULTANT INVERSE.
C                    N - ORDER OF MATRIX A
C                 BIGA - RESULTANT DETERMINANT
C                    L - WORK VECTOR OF LENGTH N
C                    M - WORK VECTOR OF LENGTH N
C
              NK=-N
              DO 190 K=1,N
              NK=NK+N
              L(K)=K
              M(K)=K
              KK=NK+K
              BIGA=A(KK)
              DO 30 J=K,N
              IZ=N*(J-1)
              DO 30 I=K,N
              IJ=IZ+I
           10 IF(DABS(BIGA)-DABS(A(IJ))) 20,30,30
           20 BIGA=A(IJ)
              L(K)=I
              M(K)=J
           30 CONTINUE
C
C                INTERCHANGE ROWS
C
              J=L(K)
              IF(J-K) 60,60,40
           40 KI=K-N
              DO 50 I=1,N
              KI=KI+N
              HOLD=-A(KI)
              JI=KI-K+J
              A(KI)=A(JI)
           50 A(JI) =HOLD
C
C                INTERCHANGE COLUMNS
C
           60 I=M(K)
              IF(I-K) 90,90,70
           70 JP=N*(I-1)
              DO 80 J=1,N
              JK=NK+J
              JI=JP+J
              HOLD=-A(JK)
              A(JK)=A(JI)
           80 A(JI) =HOLD
C
C                DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C                CONTAINED IN BIGA)
C
           90 IF(BIGA) 110,100,110
          100 RETURN
```

```
      110 DO 130 I=1,N
          IF(I-K) 120,130,120
      120 IK=NK+I
          A(IK)=A(IK)/(-BIGA)
      130 CONTINUE
    C
    C        REDUCE MATRIX
    C
          DO 160 I=1,N
          IK=NK+I
          IJ=I-N
          DO 160 J=1,N
          IJ=IJ+N
          IF(I-K) 140,160,140
      140 IF(J-K) 150,160,150
      150 KJ=IJ-I+K
          A(IJ)=A(IK)*A(KJ)+A(IJ)
      160 CONTINUE
    C
    C        DIVIDE ROW BY PIVOT
    C
          KJ=K-N
          DO 180 J=1,N
          KJ=KJ+N
          IF(J-K) 170,180,170
      170 A(KJ)=A(KJ)/BIGA
      180 CONTINUE
    C
    C        REPLACE PIVOT BY RECIPROCAL
    C
          A(KK)=1.0/BIGA
      190 CONTINUE
    C
    C        FINAL ROW AND COLUMN INTERCHANGE
    C
          K=N
      200 K=(K-1)
          IF(K) 270,270,210
      210 I=L(K)
          IF(I-K) 240,240,220
      220 JQ=N*(K-1)
          JR=N*(I-1)
          DO 230 J=1,N
          JK=JQ+J
          HOLD=A(JK)
          JI=JR+J
          A(JK)=-A(JI)
      230 A(JI) =HOLD
      240 J=M(K)
          IF(J-K) 200,200,250
      250 KI=K-N
          DO 260 I=1,N
          KI=KI+N
          HOLD=A(KI)
          JI=KI-K+J
          A(KI)=-A(JI)
      260 A(JI) =HOLD
          GO TO 200
```



```
      270 RETURN
          END
```